APPLICATION

FOR

UNITED STATES LETTERS PATENT

APPLICANT NAME: S. M. Eustis et al.

TITLE: SELF-TEST ARCHITECTURE TO IMPLEMENT DATA COLUMN REDUNDANCY IN A RAM

DOCKET NO.: BUR920030049US1

INTERNATIONAL BUSINESS MACHINES CORPORATION

Certificate of Mailing Under 37 CFR 1.10

I hereby certify that, on the date shown below, this correspondence is being deposited with the United States Postal Service in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 as "Express Mail Post Office to Addressee"

"Express Mail" Label No.: EU133644831US

On: 9/09/03

June M. Mitchell

Typed or Printed Name of Person Mailing Correspondence

June M. Mitchell 9/09/03

Signature **Date**

SELF-TEST ARCHITECTURE TO IMPLEMENT DATA COLUMN REDUNDANCY IN A RAM

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] The present invention relates generally to self-test architectures to implement data column redundancy in a Random Access Memory (RAM), either a Dynamic RAM (DRAM) or a Static RAM (SRAM), and is particularly applicable to compileable memories which are memories designed and compiled from basic building blocks termed kernels. The present invention is also particularly applicable to embedded RAM (eRAM) within microprocessor or logic chips, although it also has applicability to stand alone RAM chips.

[0002] More particularly, the subject invention pertains to self-test architectures to implement data column redundancy in a RAM memory that uses column and row redundancy with a totally integrated self-test and repair capability. A specific embodiment of a compileable 1-port SRAM is disclosed, although as noted the present invention has wider applicability to RAM memories in general, and in SRAM memories has wider applicability to dual-port or multi-port SRAM memories.

2. Discussion of the Prior Art

[0003] Compileable memories are memories which are custom designed and compiled from basic building blocks termed kernels. Compileable memories are designed or specified to have designated parameters of number of words, word width, and number of bitlines. With those parameters being specified, the design of the compileable memory is then assembled from the building block kernels.

[0004] The prior art has used many implementations of redundancy in embedded memories using both spare rows and spare columns. When using both row and column redundancy, a circuit designer must decide how best to implement the repair solution, such that the repair makes optimum use of the spare rows and columns. Some compileable memories have offered spare columns, but have offered no support for self-test, redundancy allocation or steering multiplexors. These memories simply compile an additional data column. For example, if a customer orders a 128K x 32 memory, the compiler actually gives

the customer a 128K x 33 memory. The customer then has to figure out how to test and find the bad column, and also has to implement the steering logic as well as the fusing support circuits.

[0005] The prior art has also provided recent developments within ASICs to include row redundancy in compileable memories.

SUMMARY OF THE INVENTION

[0006] The present invention relates generally to self-test architectures to implement data column redundancy in a Random Access Memory (RAM), either a Dynamic RAM (DRAM) or a Static RAM (SRAM), and is particularly applicable to compileable memories and to embedded RAM (eRAM) within microprocessor or logic chips. Two passes of self-test are used on a memory. The first pass of self-test determines the worst failing column, the column with the largest number of unique failing row addresses, after which the spare column is allocated to replace the worst failing column. In the second pass of self-test, the BIST (Built In Self-Test) collects unique failing row addresses. At the completion of the second pass of self-test, the spare rows are allocated. Once the second pass of self-test is completed, the column and unique failing row addresses are transported to the e-fuse macros and permanently stored in the chip.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The foregoing objects and advantages of the present invention for self-test architecture to implement data column redundancy in a RAM may be more readily understood by one skilled in the art with reference being had to the following detailed description of several embodiments thereof, taken in conjunction with the accompanying drawings wherein like elements are designated by identical reference numerals throughout the several views, and in which:

[0008] Figure 1 illustrates the approach of the present invention to determine the worst failing column in the memory in a first pass of self-test, which involves testing one column at a time while counting the number of unique failing rows.

[0009] Figure 2 illustrates further details of a section or bit slice of the decoder for column mask or redundancy select block shown in Figure 1, which section or bit slice would be provided for each of the columns of the memory.

[0010] Figure 3 illustrates a logic flow diagram of the self-test procedure of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0011] The present invention provides a new approach to self-testing, allocating and repairing RAM memories and particularly compileable memories using both spare rows and columns.

[0012] The new approach of the subject invention uses two passes of self-test of a memory. The first pass of self-test determines the worst failing column, the column with the largest number of unique failing row addresses. At the completion of the first pass of self-test, the spare column is allocated to replace the worst failing column. In the second pass of self-test, the BIST (Built In Self-Test) collects unique failing row addresses as it does today for memories with spare rows only. At the completion of the second pass of self-test, the spare rows are then allocated. Once the second pass of self-test is completed, the column and unique failing row addresses are transported to the e-fuse macros and permanently stored in the chip.

[0013] Figure 1 illustrates the approach of the present invention to determine the worst failing column in the memory in a first pass of self-test, which involves testing one column at a time while counting the number of unique failing row addresses in that column. Figure 1 illustrates a compileable 1-port SRAM memory having a number of data columns 0, 1...n, and a redundant data column. A decoder 10 (for column mask or redundancy select) is provided to select one of the data columns, including the redundant data column, and to mask all remaining columns.

[0014] The BIST generates a column address signal designating a particular column in memory to be selected and tested, which is input to a first register 11. During BIST

column testing, the first register outputs the column address signal through a multiplexer 12 to the decoder 10 to select the particular column to be tested. The first register 11 also outputs a worst tested column address signal to a second repair register 13, and in normal operation, the second repair register 13 outputs the stored worst column address signal through the multiplexer 12 which is decoded by the same decoder 10 to implement the redundant data column in place of the worst tested column.

[0015] During the first pass of self-test, all columns except the selected tested column are masked or deselected such that only the un-masked or selected column can produce an error in a data-out comparator 14. The pass/fail signal from the dataout comparator 14 is then used to enable the Failing Address Register (FAR) 15 for the selected data column to store each unique failing row address, provided it has not already been stored. The FAR register includes a register entry for row address 0, row address 1...row address n, as indicated schematically in Figure 1. As the unique failing row addresses are stored, a failing row counter 16 is enabled through an OR gate 17 to count the number of unique failing row addresses for the unmasked column.

[0016] At the end of testing of the unmasked column, the count value in the counter 16 for the unmasked column is compared in comparator 18 with a worst count value from previously tested columns stored in an error count register 19. If the count value in the counter 16 for the unmasked column exceeds the previously stored worst count value from previously tested columns, then the unmasked column is determined to be the worst column tested so far. The new count value for the unmasked register is then stored in the error count register 19, and the bit-address of the unmasked data column is stored in the repair register 13. The bit-address is the binary value sent from the self-test circuit to the decoder, such that the decoder's output provides a means to identify and select the un-masked column. The stored count value in register 19 is subsequently compared to the next column's count value after testing is completed on the next column. After testing of each column, the FAR 15 and failing row counter 16 are cleared before testing the next column. At the completion of testing of all columns individually, the worst column's bit-address is stored and saved.

[0017] In an alternative embodiment, a circuit could also be implemented that does not allow the use of the spare column unless the worst column's count value exceeds a given threshold value, for example 2.

[0018] In embodiments implemented in wide RAMs, the RAM can be divided into sections of adjacent columns, with each section having its own redundant column to replace a worst column in that section, and each section can be tested in parallel with other sections of columns.

[0019] In some embodiments, when the number of unique failing row addresses in two columns exceeds the number of redundant rows in the RAM, the RAM is designated as unrepairable. For instance, a flag can be set when the number of unique failing rows in a given column exceeds the number of redundant rows in the RAM, and if a second column exceeds the number of redundant rows in the RAM, and the flag is already set, then the RAM is designated as unrepairable.

[0020] Figure 2 illustrates further details of a section or bit slice of the decoder for column mask or redundancy select block shown in Figure 1, which section or bit slice would be provided for each of the columns of the memory. The bit bus 20 is connected to an 8 bit decoder 21 which comprises first and second NAND gates 22, 23 coupled through a NOR gate 24 which produces an output BADBIT(i), wherein (i) refers to a column designated (i), and (i-1) refers to the immediately preceding column. The output BADBIT(i) is input to an OR gate 25, which receives a second input MUXSEL (i-1), and produces an output to a first AND gate 26. An input TESTMODE, which indicates a test mode as opposed to normal operation, is input to a third NAND gate 27, which also receives the BADBIT(i) signal inverted by an inverter 28 as an input, and produces a MASKN(i) output which serves to mask out untested bits. The TESTMODE signal is also inverted by inverter 29 and the inverted signal is a second input to the first AND gate 26, which produces an output MUXSEL(i) signal to select and control the redundancy mutiplexers provided for the memory.

[0021] The stored bit-address value stored in the repair register 13 at the end of the first pass of self-test is then used to enable the spare column, prior to the second pass of the self-test. The same decoder 10 that was used to mask/unmask is then also used to select the steering multiplexors for implementing the spare column, as illustrated in Figure 2.

[0022] Memory cells within the spare column are tested during the second pass during which the FAR is used to collect failing row addresses. At the end of the second pass of self-test, the FAR values are used to allocate and implement the spare rows.

[0023] Figure 3 illustrates a logic flow diagram of the self-test procedure of the present invention. The system enters the column test mode at 30, indicated by the signal TESTMODE of Figure 2, and the system proceeds to test the first column at 32, counting the number of unique failing row addresses per column. In the column being tested, each next row address in the column is tested at 34, and at 36 the address of each failed new wordline is loaded into the FAR, and the counter 16 which counts the number of unique failing row addresses is incremented at 38, and the next address in the column is tested until the last address in the column is tested at 40. At 42, if the total number of fails counted by the counter 16 for the tested column exceeds the recorded previous maximum number of fails, then at 44 the column address and number of fails of the tested column are stored in respectively the repair register 13 and the register 19. If at 42 the total number of fails for the tested column does not exceed the recorded previous maximum number of fails, then at 46 the system determines if all columns have been tested. If not, the FAR register is reset at 48 and the column fail counter 16 is reset at 50, and testing of the next column begins at 32. If testing of all columns is completed at 46, the system implements the worst column at 52, substituting the redundant data column for the data column register having the most fails.

[0024] The system then exits the first pass of self-test, the column test mode, and proceeds at 54 to the second pass of self-test, to test the memory with usual row test patterns, after which at 56 the results of the column and row testing are permanently implemented and recorded by blowing fuses corresponding to the test results.

[0025] While several embodiments and variations of the present invention for self-test architecture to implement data column redundancy in a RAM are described in detail herein, it should be apparent that the disclosure and teachings of the present invention will suggest many alternative designs to those skilled in the art.